

Adjusting Resource Limits for Login Sessions and PBS Jobs

There are several settings on the front-end systems (PFEs) and compute nodes that define hard and soft limits to control the amount of resources allowed per user, or per user process. You can change the default soft limits for your sessions, up to the maximum hard limits, as needed.

Keep in mind that there might be different default limits set for an SSH login session and a PBS session, and among the various front-end systems and compute nodes.

Viewing Soft and Hard Limits

To view the soft limit settings in a session, run:

- For csh: `% limit`
- For bash: `$ ulimit -a`

The output samples below show soft limit settings on a PFE.

For csh:

```
% limit
cputime      unlimited
filesize     unlimited
datasize     32700000 kbytes
stacksize    unlimited
coredumpsize unlimited
memoryuse    32700000 kbytes
vmemoryuse   unlimited
descriptors  1024
memorylocked unlimited
maxproc      400
maxlocks     unlimited
maxsignal    255328
maxmessage   819200
maxnice      0
maxrtprio    0
maxrttime    unlimited
```

For bash:

```
$ ulimit -a
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) 32700000
scheduling priority     (-e) 0
file size               (blocks, -f) unlimited
pending signals         (-i) 255328
max locked memory       (kbytes, -l) unlimited
max memory size         (kbytes, -m) 32700000
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
real-time priority      (-r) 0
stack size              (kbytes, -s) unlimited
cpu time                (seconds, -t) unlimited
max user processes      (-u) 400
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

To view the hard limits, run:

- For csh: `% limit -h`
- For bash: `$ ulimit -Ha`

Modifying Soft Limits

To modify the soft limit setting of a resource, type the name of the resource after **limit** (for **cs**) or type the option after **ulimit** (for **bash**). For example, to reset the **stacksize** limit to 200,000 KB, run:

- For **cs**: `% limit stacksize 200000`
- For **bash**: `$ ulimit -Ss 200000`

Common Issues with the Default Limits

The following list describes a few common issues with default limits.

- **stacksize** (stack size)

On the PFEs, the **stacksize** soft limit is set to **unlimited**. Some plotting packages, such as Tecplot, require a small stack size to function properly, so you will have to decrease the **stacksize** in order to use them. However, if the **stacksize** is set too small when applications are run, segmentation faults (segfaults) commonly occur. If you run plotting packages and your own applications in the same session, you will need to alternate between a small and a large stack size accordingly.

- **maxproc** (maximum user processes)

The **maxproc** limits are set to 400 (soft) and 600 (hard) on the PFEs. On the compute nodes, the limits are set to more than 100,000 (the size varies by node type).

If you encounter one of the following error messages on the PFEs, this indicates that you have reached the **maxproc** soft limit:

```
can't fork process
or
fork: Resource temporarily unavailable
```

This may happen during compilation, especially if you run a parallel **make** operation using **make -j**.

Note: Each Tecplot session can consume more than 100 user processes. If you run multiple instances of Tecplot on the same PFE, you might reach the **maxproc** soft limit.

- **memoryuse** and **vmemoryuse** (maximum memory size and virtual memory)

The soft limits for **memoryuse** and **vmemoryuse** are set to **unlimited** on most compute nodes. The exceptions are the PFEs, where **memoryuse** is set to 32 GB.

The **vmemoryuse** setting affects the outcome when you try to allocate memoryâfor example, when you use the **allocate** command as follows:

```
allocate (a(i),stat=ierror)
```

The **memoryuse** setting affects the referencing of array **a**. Keep in mind that running your applications on nodes with different settings will result in different behaviors.

- **coredumpsize** (core file size)

Core files are useful for debugging your application. However, large core files may consume too much of your disk quota. In addition, the generation of large core files in a PBS job could occupy the buffer cache of the nodes, waiting to be flushed to disk. If the

buffer cache on the nodes is not flushed by the PBS epilogue of the current job and the PBS prologue of the next job, the nodes will be unusable.

Article ID: 553

Last updated: 13 May, 2021

Revision: 32

Running Jobs with PBS -> Optimizing/Troubleshooting -> Adjusting Resource Limits for Login Sessions and PBS Jobs
<https://www.nas.nasa.gov/hecc/support/kb/entry/553/>